5

FACILITATING MINIMUM SPACING AND/OR WIDTH CONTROL DURING OPTICAL PROXIMITY CORRECTION

Inventors: Youping Zhang and Christophe Pierrat

15

BACKGROUND

20 **Field of the Invention**

The invention relates to the process of designing and fabricating semiconductor chips. More specifically, the invention relates to a method and an apparatus that facilitates minimum spacing and/or width control during an optical proximity correction process for a mask that is used in manufacturing an

25 integrated circuit.

Related Art

Recent advances in integrated circuit technology have largely been accomplished by decreasing the feature size of circuit elements on a

semiconductor chip. As the feature size of these circuit elements continues to decrease, circuit designers are forced to deal with problems that arise as a consequence of the optical lithography process that is typically used to manufacture integrated circuits. This optical lithography process begins with the

5   formation of a photoresist layer on the surface of a semiconductor wafer. A mask composed of opaque regions, which are formed of chrome, and light-transmissive clear regions, which are generally formed of quartz, is then positioned over this photo resist layer coated wafer. (Note that the term "mask" as used in this specification is meant to include the term "reticle.") Light is then shone on the

10   mask from a visible light source, an ultraviolet light source, or more generally some other type of electromagnetic radiation together with suitably adapted masks and lithography equipment.

This light is reduced and focused through an optical system that contains a number of lenses, filters and mirrors. The light passes through the clear regions of

15   the mask and exposes the underlying photoresist layer. At the same time, the light is blocked by opaque regions of the mask, leaving underlying portions of the photoresist layer unexposed.

The exposed photoresist layer is then developed, through chemical removal of either the exposed or non-exposed regions of the photoresist layer.

20   The end result is a semiconductor wafer with a photoresist layer having a desired pattern. This pattern can then be used for etching underlying regions of the wafer.

One problem that arises during the optical lithography process is "line end shortening" and "pullback". For example, the upper portion of FIG. 1 illustrates a design of a transistor with a polysilicon line 102, running from left to right, that

25   forms a gate region used to electrically couple an upper diffusion region with a lower diffusion region. The lower portion of FIG. 1 illustrates a printed image that results from the design. Note that polysilicon line 102 has been narrowed

2

using optical phase shifting in order to improve the performance of the transistor by reducing the resistance through the gate region.

Also note that because of optical effects and resist pullback there is a significant amount of line end shortening. This line end shortening is due to

5 optical effects that cause the light to expose more of the resist under a line end than under other portions of the line.

In order to compensate for line end shortening, designers often add additional features, such as "hammer heads," onto line ends (see top portion of FIG. 2). The upper portion of FIG. 2 illustrates a transistor with a polysilicon line

10 202, running from left to right, which forms a gate region used to electrically couple an upper diffusion region with a lower diffusion region. A hammer head 204 is included on the end of polysilicon line 202 to compensate for the line end shortening. As is illustrated in the bottom portion of FIG. 2, these additional features can effectively compensate for line end shortening in some situations.

15 These additional features are typically added to a layout automatically during a process known as optical proximity correction (OPC). For example, FIG. 3 illustrates line end geometry 302 (solid line) prior to OPC and the resulting corrected line end geometry 304 after OPC (dashed line). Note that the corrected line end geometry 304 includes regions with a positive edge bias in which the size

20 of the original geometry 302 is increased, as well as regions of negative edge bias in which the size of the original geometry 302 is decreased.

Unfortunately, the OPC process for a given edge typically does not take into account the amount of correction of neighboring edges, which can be located in the same feature or in different features. For example, a positive bias for a

25 given edge can possibly cause a violation of a minimum spacing requirement between the given edge and an edge of a neighboring feature. In another example, a negative bias for a given edge can cause a violation of a minimum width

3

requirement between the given edge and an opposing edge of the same geometric feature (see FIG. 9).

Undesirable effects caused by these interactions can be mitigated through a subsequent design rule checking (DRC) operation that takes place after the OPC

5   process. However, this subsequent DRC operation takes place without the benefit of information that is available during the OPC process. Hence, modifications made by the subsequent DRC operation may adversely effect corrections made during the OPC process. For example, in FIG. 4A, an OPC process causes hammerheads 404 and 414 to be added to line ends 402 and 412, respectively. A

10   subsequent DRC operation detects a minimum spacing violation between hammerheads 404 and 414, and corrects this violation by cutting away portions 406 and 416 from hammerheads 404 and 414. However, these corrections also adversely affect the optical proximity correction.

Undesirable effects caused by these interactions can also be mitigated

15   during the OPC process. In some systems, when the OPC process detects a spacing constraint violation the OPC process simply stops and does not add an optical proximity correction or adds a scaled back optical proximity correction. For example, referring to FIG. 4B, the OPC process successfully places a hammerhead 404 on line 402. However, when the OPC process subsequently

20   attempts to place a hammerhead on line end 412 a minimum spacing violation is detected, and no hammerhead is placed since there is not enough room for the correction. Note that this outcome results in an undesirable asymmetry between the resulting images of line ends 402 and 412.

What is needed is a method and an apparatus that facilitates minimum

25   spacing and/or width control without adversely affecting corrections made by the optical proximity correction process.

4

SUMMARY

One embodiment of the invention provides a system that facilitates minimum spacing and/or width control during an optical proximity correction operation for a layout of a mask used in manufacturing an integrated circuit.

5   During operation, the system considers a target edge of a first feature on the mask and then identifies a set of interacting edges in proximity to the target edge. Next, the system performs the optical proximity correction operation, wherein performing the optical proximity correction operation involves applying a first edge bias to the target edge to compensate for optical effects in a resulting image

10   of the target edge. While applying the first edge bias to the target edge, the system allocates an available bias between the first edge bias for the target edge and a second edge bias for at least one edge in the set of interacting edges.

In a variation on this embodiment, applying the first edge bias can involve adding a positive edge bias that increases the size of the first feature or adding a

15   negative edge bias that decreases the size of the first feature.

In a variation on this embodiment, the second edge belongs to a second feature so that the distance between the target edge and the second edge defines a distance between the first feature and the second feature. In this variation, applying the first edge bias to the target edge involves satisfying a minimum

20   spacing requirement between the target edge and the second edge. In a further variation, applying the first edge bias to the target edge additionally involves satisfying a minimum width requirement between the target edge and an opposing edge of the first feature.

In a variation on this embodiment, the second edge is also an edge of the

25   first feature so that a distance between the target edge and the opposing edge defines a distance across a gap between portions of the first feature.

5

In a variation on this embodiment, the second edge is an opposing edge of the first feature so that a distance between the target edge and the opposing edge defines a width of the first feature. In this variation, applying the first edge bias to the target edge involves satisfying a minimum width requirement for the first

5 feature between the target edge and the second edge.

In a variation on this embodiment, applying the first edge bias to the target edge involves considering an edge type of the target edge and considering an edge type of the second edge.

In a variation on this embodiment, allocating the available bias between

10 the target edge and the second edge involves ensuring that the first edge bias of the target edge satisfies a minimum spacing requirement between the target edge and each edge in the set of interacting edges.

In a variation on this embodiment, allocating the available bias between the target edge and the second edge involves ensuring that the first edge bias of

15 the target edge satisfies a minimum width requirement between the target edge and each edge in the set of interacting edges.

In a variation on this embodiment, the available bias is allocated based on relative weights assigned to the target edge and the second edge.

In a variation on this embodiment, allocating the available bias involves

20 iteratively updating bias allocated to the target edge and the second edge in a manner that satisfies minimum spacing requirements or minimum width requirements.

BRIEF DESCRIPTION OF THE FIGURES

25 FIG. 1 illustrates the line end shortening problem.

FIG. 2 illustrates the use of a hammerhead to compensate for the line end shortening problem.

6

FIG. 3 illustrates line end geometry prior to and after optical proximity correction.

FIG. 4A illustrates how design rule checking is applied to ensure a minimum separation.

5   FIG. 4B illustrates how a spacing constraint can stop edge movement during OPC.

FIG. 4C illustrates how minimum separation is achieved without adversely affecting optical proximity correction in accordance with an embodiment of the invention.

10   FIG. 5 is a flow chart illustrating the wafer fabrication process in accordance with an embodiment of the invention.

FIG. 6 illustrates the process of creating a mask to be used in fabricating an integrated circuit in accordance with an embodiment of the invention.

FIG. 7 is a flowchart illustrating the optical proximity correction process

15   in accordance with an embodiment of the invention.

FIG. 8 is a flow chart illustrating the process of considering spacing constraints and width constraints during the optical proximity correction process in accordance with an embodiment of the invention.

FIG. 9 illustrates minimum width and minimum spacing requirements.

20   FIG. 10 illustrates an exemplary bias computation in accordance with an embodiment of the invention.

FIG. 11 illustrates another exemplary bias computation in accordance with an embodiment of the invention.

FIG. 12 illustrates yet another exemplary bias computation in accordance

25   with an embodiment of the invention.

FIG. 13 illustrates a set of relevant edges an embodiment of the invention.

7

FIG. 14 illustrates a two-edge interaction in accordance with an embodiment of the invention.

FIG. 15 illustrates another two-edge interaction in accordance with an embodiment of the invention.

5   FIG. 16 illustrates a multi-edge interaction in accordance with an embodiment of the invention.

FIG. 17 illustrates a biasing example that depends upon the type of geometries involved in accordance with an embodiment of the invention.

FIG. 18 illustrates a biasing example that depends upon shape information

10   in accordance with an embodiment of the invention.

FIG. 19 illustrates a minimum spacing requirement across a gap between portions of the same feature in accordance with an embodiment of the invention.

FIG. 20 illustrates a situation in which both a minimum spacing requirement and a minimum width requirement must be satisfied in accordance

15   with an embodiment of the invention.

DETAILED DESCRIPTION

Wafer Fabrication Process

20   FIG. 5 is a flow chart illustrating the wafer fabrication process in accordance with an embodiment of the invention. The system starts by applying a photoresist layer to the top surface of a wafer (step 502). Next, the system bakes the photoresist layer (step 504). The system then positions a mask over the photoresist layer (step 506), and then exposes the photoresist layer through the

25   mask (step 508). Next, the system optionally bakes the wafer again (step 514) before developing the photoresist layer (step 516). Next, either a chemical etching or ion implantation step takes place (step 518) before the photoresist layer is

8

removed (step 520). (Note that in the case of a lift-off process, a deposition can take place.) Finally, a new layer of material can be added and the process can be repeated for the new layer (step 522).

5   **Design Process**

FIG. 6 illustrates the process of creating a mask to be used in the wafer fabrication process described above in accordance with an embodiment of the invention. The process starts when a circuit designer produces a design 602 in VHDL, or some other hardware description language. VHDL is an acronym for

10   VHSIC Hardware Description Language. (VHSIC is a Department of Defense acronym that stands for very high-speed integrated circuits.) The VHDL standard has been codified in Institute for Electrical and Electronic Engineers (IEEE) standard 1076-1993.

Layout 602 then feeds through a layout system 603 that performs a number

15   of functions, such as synthesis 604, placement and routing 606 and verification 608. The result is an integrated circuit (IC) layout 610, which is in the form of a hierarchical specification expressed in a format such as GDSII.

IC layout 610 then passes into OPC post-processing system 611, which performs optical proximity corrections to compensate for proximity (can be

20   optical or other) effects during the photolithography process.

The output of OPC post-processing system 611 is a new IC layout 618. New IC layout 618 subsequently passes into mask fabrication and inspection processes 620.

25   **Optical Proximity Correction**

FIG. 7 is a flowchart illustrating an optical proximity correction process which takes place in box 611 of FIG. 6 in accordance with an embodiment of the

9

invention. The process starts when the system, such as a computer system, receives an circuit layout requiring optical proximity correction (OPC) (step 702). This circuit layout can be received in a standard format, such as GDSII format, mask electronic beam exposure system (MEBES) format, and/or some other

5   format. The input circuit layout can be received by the system over a network, such as a local area network, the Internet, a network attached storage network or from storage local to the system, such as a hard drive, CD-ROM, or DVD.

Next, the system receives OPC parameters for evaluating the input circuit layout (step 704). In one embodiment of the invention, the system receives OPC

10   parameters from a user through a graphical user interface (GUI). In another embodiment, the OPC parameters are received in a data file, such as a text file or an XML file.

The system then dissects the polygon edges into OPC segments at dissection points (step 706). This is done in accordance with the OPC parameters.

15   For example, the parameters might specify that segments are to be dissected and evaluated every 120 nm, but that segments on inside corners are to be dissected every 60 nm. Next, the system places evaluation points for each OPC segment (step 708). In another embodiment, the evaluation and dissection point selection approach described in United States Patent Application No. 09/676,356 entitled

20   "Selection of Evaluation Point Locations Based on Proximity Effects Model Amplitudes for Correcting Proximity Effects in a Fabrication Layout" having inventors Youping Zhang et al., filed 29 Sep 2000 and assigned to the assignee of the present application is used at step 708.

The system then selects a segment for evaluation (step 710).

25   The process then proceeds in a double loop. The inner loop evaluates each segment once and decides whether or not to perform a correction (steps 710-718),

10

and the outer loop monitors the overall process and limits the computational time (steps 710-724). These steps are described in more detail below.

At step 710, a segment is selected. Next, the system calculates a critical dimension (CD) error at the evaluation point on the selected segment (step 712). This can be done by evaluating the optical model of the lithography process at the evaluation point. The system then determines whether to correct the segment on this iteration through the inner loop (step 714).

If the system decides to correct the segment on this iteration at step 714, the system calculates the edge bias to apply on this iteration (step 716). Note that the system considers spacing constraints in calculating the edge bias in a process that is described in more detail below with reference to FIG. 8. If no correction is be performed in this iteration, the process continues at step 718. Note that the amount of bias that can be applied to an edge segment in a single iteration may be limited according to the OPC parameters received at step 704. For example, a single edge may be outwardly biased on one iteration, e.g. + 5 nm and then further biased on a subsequent iteration.

At step 718, if there are more segments to process in this iteration of the inner loop, the process continues at step 710. Otherwise, the system proceeds to update the layout with the edge biases that have been calculated for each segment (step 720).

Finally, the system determines if stopping criteria have been satisfied (step 724). If not, the system returns to step 710 for another iteration. Otherwise, the process ends. The stopping criteria can include, reaching a preset maximum number of iterations, all segments reaching a maximum correction, exhaustion of allowed computational time, all segments being fully corrected, some segments are determined to be non-correctable or not further correctable for certain reason (.e.g, low contrast), and/or combinations of one or more stopping criterion.

Adjusting Edge Biases Based Upon Spacing/Width Constraints

FIG. 8 is a flow chart illustrating the process of adjusting edge biases based upon spacing constraints and width constraints in accordance with an

5   embodiment of the invention. The system starts by receiving a current edge with an associated bias (step 801). Next, the system determines the shape that the current edge belongs to (step 802).

The system then gets the next relevant edge/shape that shares spacing with the current edge (step 804). Relevant edges/shapes are those that share immediate

10   space and width with the current edge under consideration, i.e., any edges whose movement could potentially result in spacing violation (see FIG. 13). Relevant edges can be determined based on distance and the type of neighboring edge. The system then looks up the minimum spacing requirements between the shapes (step 806), and calculates an available positive bias between the edges (step 808).

15   The system adds this positive bias to a bank of positive biases (step 810). The system then determines if there are remaining edges (step 812). If so, the system repeats steps 804, 806, 808 and 810 for all relevant edges with spacing interactions.

If not, the system determines the minimum of all available positive biases

20   (step 814) and uses this as the maximum outward bias (see FIG. 9).

The system subsequently gets the next edge/shape that shares width with the current edge (step 816). The system looks up the minimum width requirement between the interacting shapes (step 818), and then calculates available negative bias between the edges (step 820). The system adds this negative bias to a bank of

25   negative biases (step 822). The system then determines if there are remaining edges (step 824). If so, the system repeats steps 816, 818, 822 and 822 for all relevant edges.

12

If not, the system determines the minimum of all available negative biases (step 826) and uses this as the maximum inward bias (see FIG. 9).

Although FIG. 8 is described as taking place within step 716 in FIG. 7, it does not necessarily have to happen at this step. The process described in FIG 8 is

5   an independent step that determines maximum bias values, and may take place anywhere before step 716 and after step 720 in FIG. 7 if it takes place in the loop.

Note that the above-described process can satisfy spacing constraints without adversely affecting optical proximity correction. For example, referring the FIG. 4C, an "available bias" that remains after considering the minimum

10   spacing constraint is allocated between line ends 402 and 412. Unlike the example in FIG. 4A, hammerheads 404 and 406 are not simply chopped off during a subsequent DRC operation. The resulting hammerheads 408 and 418 are instead modified to be larger in the vertical direction to compensate for a reduction of size the horizontal direction. This modification achieves a similar

15   level of optical proximity correction while satisfying the minimum spacing requirement between line ends 402 and 412.

Furthermore, unlike the example illustrated in FIG. 4B, the optical proximity correction is allocated symmetrically between line ends 402 and 412. (Although asymmetric allocations are also possible depending upon the

20   requirements of the situation.)

Examples

FIG. 10 illustrates an exemplary bias correction in accordance with an embodiment of the invention. In this example, an edge of shape A 1002 is in

25   close proximity to an edge of shape B 1004. In this case, the system looks up a minimum spacing, s_{min}, between shape A 1002 and shape B 1004. Recall that s_{min} can depend upon the types of shapes that are involved.

13

Next, the system performs an iterative process to produce the correction. For each iteration, i, the system considers an existing correction 1003 for shape A 1002, and an existing correction 1005 for shape B 1004. The system then calculates a maximum update, $a(i)_{max}$, for correction 1003, and a maximum update, $b(i)_{max}$, for correction 105. For example, these maximum updates can be computed using the following formulas:

$$a(i)_{max}=(s(i)- s_{min}) \times r_a/(r_a+r_b); \text{ and}$$
$$b(i)_{max}=(s(i)- s_{min}) \times r_b/(r_a+r_b).$$

Note that these formulas make use of coefficients r_a and r_b that specify a relative weighting for corrections between shape A 1002 and shape B 1004. These coefficients r_a and r_b can depend upon shape information.

The ratios are used to compromise between the two feature's available spaces for corrections. Note that only the relative value of r_a and r_b, $r_a:r_b$, is important, not their absolute values. The simplest case is to choose $r_a=1$ and $r_b=1$, i.e., shape A and shape B split the available spacing at all time. In one implementation, the ratios are initially chosen so that they favor those features whose accuracy is relatively more important over those less important. The ratios then gradually migrate toward favoring the features that need further corrections.

Consider an example. Suppose shape A is an edge, shape B is a line end, and we favor edges over line ends. Then the initial values of the ratios may be selected as, say $r_a=4$, $r_b=1$, i.e., for every 5nm of available spacing, 4nm is assigned to the edge and 1nm is assigned to the line end. As the correction continues, it turns out that the edge quickly meets its spec by proper biasing while the line end is still missing the spec due to insufficient biasing due to spacing constraints. When there is still spacing left to be assigned, the favor turns toward the line end, e.g., $r_a=1$, $r_b=4$. One such formula, for example, can be as follows:

14

$$r_a(i) = \lambda_a(i)r_a(0) + (\lambda_a(0) - \lambda_a(i))\delta_a(i)$$
$$r_b(i) = \lambda_b(i)r_b(0) + (\lambda_b(0) - \lambda_b(i))\delta_b(i)$$
$$\lambda_a(i) = \lambda_a(0)e^{-k_a i}$$
$$\lambda_b(i) = \lambda_b(0)e^{-k_b i}$$

where $r_a(0)$, $r_b(0)$ are provided by user to specify initial preferences, and $\delta_a(i)$, $\delta_b(i)$ represent how far the critical dimensions are from spec or the assigned biases are from the asked biases (i.e., biases would have been applied without spacing constraint). The coefficient $\lambda_a(i)$, $\lambda_b(i)$ are weighted toward initial assignments $r_a(0)$, and $r_b(0)$, and gradually migrate toward needs (until finally they completely depend on needs).

There could also be cases where a user absolutely requires edges meet specifications before considering line end. This can be treated by using the same formula, where we simply set $r_a(0) >> r_b(0)$ and $\lambda_a(0) >> \lambda_b(0)$.

These maximum updates, $a(i)_{max}$ and $b(i)_{max}$, are then used to update corrections 1003 and 1005, respectively. The process is repeated until either, all of the necessary corrections are made, a correction limit is reached, or an iteration limit is reached.

FIG. 11 illustrates another exemplary bias correction in accordance with an embodiment of the invention. In this example, an edge a interacts with multiple edges, b, c, d, and e. In this case, the system determines an available bias for each of the four interactions, and selects the smallest of these as the maximum outward correction to constraint the available biasing for edge a. In the case of edges a and b, the two shapes are not facing each other like in FIG 10. Determining available bias has to take into account the effects of corners. For example, the available bias between edge a and edge b can be computed using the following distance formula:

15

available_bias(a,b) = OPC maximum adjustment parameter if $d_{ab} \geq s_{min}$;
and

$$\sqrt{s_{ab} - \sqrt{s_{min}^{2} - d_{ab}^{2}}} \; \frac{r_a}{r_a + r_b} \quad \text{if } d_{ab} < s_{min}.$$

5   Note that s_{ab} represents a horizontal distance between edges a and b, and d_{ab} represents a vertical distance between edges a and b. Furthermore, note that the available bias for the other three edges c, d and e can be computed similarly. This available bias is then used to update the bias for edge a in an iterative process

10   which is similar to the single interaction case described above with reference to FIG. 10.

FIG. 12 illustrates yet another exemplary bias correction in accordance with an embodiment of the invention. This example illustrates an interaction between edges of a corner 1202 and a corner 1204. Note that the optical

15   proximity correction for corner 1202 includes a vertical correction associated with a_{max} and a horizontal correction associated with b_{max}. Similarly, the optical proximity correction for corner 1204 involves a vertical correction associated with c_{max} and a horizontal correction associated with d_{max}. The corrections a_{max}, b_{max}, c_{max}, and d_{max} can be calculated by the following formulas, wherein Θ is the angle

20   between a line that connects the corners and a horizontal line:

$$b_{max} = (s(i) - s_{min}) \times r_a/(r_a + r_b) sin \; \Theta \; ;$$
$$a_{max} = (s(i) - s_{min}) \times r_a/(r_a + r_b) cos \; \Theta \; ;$$
$$d_{max} = (s(i) - s_{min}) \times r_d/(r_c + r_d) sin \; \Theta \; ; \text{ and}$$
25   $$c_{max} = (s(i) - s_{min}) \times r_c/(r_c + r_d) cos \; \Theta.$$

The quantities a_{max}, b_{max}, c_{max}, and d_{max} are then used to update the optical proximity corrections for corners 1202 and 1204. The above formula requires that

16

the angle, Θ, be maintained as a constant. This may be a desirable limitation on the application of OPC, alternatively other constraints can be used and the equations factored to account for changes in the angle, Θ.

FIG. 13 illustrates how a number of relevant edges 1301-1307 are identified from an edge under consideration 1300. To determine which edges will interact due to minimum spacing requirements, lines are projected in horizontal and vertical directions from the edge under consideration 1300 to find relevant edges 1301-1305. To determine edges that will interact due to minimum width requirements, lines are projected into the interior of the geometry from the edge under consideration 1300 to find relevant edges 1306-1307. Illustrated by dashed lines. Specifically, edges 1304 and 1305 share immediate spacing with 1300 as part of those edges project directly into 1300 (facing each other). Edges 1302 and 1303 share spacing with 1300 because the corner 1311 can grow towards 1300. Edge 1308 does not interact with edge 1300 directly as it is behind corner 1309. Similarly, edge 1310 is not relevant for computing the spacing of edge 1300. Widths are handled in a similar fashion.

Referring to FIG. 14, any two-edge interaction can be processed in the following manner. Suppose d_a and d_b are applied. Calculate the minimum distance after the correction $s = f(d_a, d_b)$. Then set $s = s_{min}$ and substitute $d_a/d_b = r_a/r_b$, and solve for d_a and d_b, which will be the available maximum correction for this particular edge interaction. For example, in the case illustrated in FIG. 15, available biases are calculated using the following equations:

$$(y_{ab} - d_a)^2 + (x_{ab} - d_b)^2 = s_{min}^2$$
$$d_a / d_b = r_a / r_b$$

For any multi-edge shape interaction, the system first determines the direction that will shorten the spacing the fastest (see FIG. 16). The system

17

assumes biasing in this worst-case direction to calculate the "worst-case" spacing value. Next, the system sets the ratio among edges in the shape to be along that direction and then calculates distance assuming biasing along the worst-case direction. More specifically, the system sets the spacing value to s_{min}, and applies the ratio to solve for the biasing along this worst-case direction. Next, the system distributes the biasing between the affected edges. Note that this process is iterative so the worst case biasing direction can change for subsequent iterations. Also note that bias computations are performed one edge at a time. At times when an edge simultaneously belongs to two corners (e.g. line end) the above operation (multi-edge shape interaction) has to be performed twice, once for each corner. Although the bias computations for an edge take into consideration interactions with other edges, biasing for these other edges is not affected by the bias computations for the edge. The technique described above can also be applied to minimum width requirements.

It can be proven that as the number of iterations grows, this method is non-conservative, i.e., the final biasing will use up the available spacing or width to achieve the best possible OPC results.

Biasing ratios can also depend upon shape information and the type of geometries involves. For example, FIG. 17 illustrates a minimum spacing requirement along the line, S, between trim region 1710 for a gate region of a transistor and a neighboring wire 1712. However, the dimensions of trim region 1710 cannot change. Hence, the biasing will only be applied to the wire 1712.

In another example illustrated in FIG. 18, there is a minimum spacing requirement along the line, S, between line end 1802 and edge 1804. In this example, biasing can applied in different ratios between line end 1802 and edge 1804 based on shape information.

18

In yet another example illustrated in FIG. 19, there is a minimum spacing requirement along the line, S, between a first edge of feature 1902 and a second edge of feature 1902 across a gap between portions of feature 1902.

In a further example illustrated in FIG. 20, there is a minimum spacing requirement to satisfy along the line, S, between an edge 2006 of feature 2002 and an edge 2008 of feature 2004. At the same time, there is a minimum width requirement to satisfy along the line, W, between edge 2006 of feature 2002 and an opposing edge 2010 of feature 2002.

The foregoing description is presented to enable one to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. Thus, the invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

The data structures and code described in this detailed description can be stored on a computer readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. This includes, but is not limited to, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs) and DVDs (digital versatile discs or digital video discs), and computer instruction signals embodied in a transmission medium (with or without a carrier wave upon which the signals are modulated). For example, the transmission medium may include a communications network, such as the Internet.

Note that the invention can be applied to any type of lithographic process for fabricating semiconductor chips, including processes that make use of, deep-

19

ultraviolet (DUV) radiation, extreme ultraviolet (EUV) radiation, X-rays, and electron beams, along with suitably modified masks.

The foregoing descriptions of embodiments of the invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the invention to the forms disclosed. Accordingly, many modifications and variations will be apparent. Additionally, the above disclosure is not intended to limit the invention. The scope of the invention is defined by the appended claims.

20